JOURNAL OF COMPUTATIONAL PHYSICS 93, 144–170 (1991)

Computer Simulations of Multiplicative Stochastic Differential Equations

P. D. DRUMMOND AND I. K. MORTIMER

Physics Department, The University of Queensland, St. Lucia Queensland 4072, Australia

Received March 6, 1989; revised January 19, 1990

A class of robust algorithms for the computer simulation of stochastic differential equations with multiplicative noise is investigated. Excellent agreement is obtained with the known analytic behaviour of the Kubo oscillator in the white noise limit. The algorithms include a known first-order one-dimensional explicit method, as well as implicit methods of increased stability. A distinction is drawn between classes of stochastic differential equations depending on the type of spatial variation or curvature defined by the diffusion tensor. This allows greatly simplified numerical implementations of the new algorithms in certain cases. The results of different techniques are compared for the case of the Kubo oscillator, where a semi-implicit technique gives the greatest accuracy.

1. INTRODUCTION

Computer simulations of stochastic differential equations [1] offer a powerful technique for obtaining information in statistical physics. Analytic techniques are often only useful in the linearised, small noise limit of nonlinear problems. This limitation can be overcome by the technique of direct computer simulation. Stochastic equations are even more interesting in view of the well-known equivalence between a Fokker–Planck equation and a stochastic equation. Since Fokker–Planck equations with large dimensionality are difficult to analyse numerically, stochastic equations offer a useful alternative technique.

In practical terms, there are relatively few well-understood numerical algorithms for these types of calculation. The numerical problem is particularly acute for stochastic equations obtained from Fokker–Planck equations, as these involve infinite-bandwidth noise sources. The theoretical work of Ito [2], Stratonovich [3], and others has defined precisely how to mathematically interpret the singular noise sources that occur. This requires some modification of standard calculus techniques in the case of Ito's approach. Thus, the mathematical properties of stochastic equations are relatively well known.

Nevertheless, the numerical simulation of these equations has not been studied with the same detail as for non-stochastic equations. In particular, the usual limit of small step size has a different behaviour in the stochastic case from that in differential equations with continuous source functions. As the step size becomes

0021-9991/91 $3.00

The quantitative numerical comparisons use the Kubo oscillator to illustrate the differences. This exactly soluble equation is one of the simplest nontrivial examples of multiplicative stochastic noise. The results show that in this case the Ito–Euler algorithm has the worst local error performance and is especially error-prone for higher order moments. A first-order explicit algorithm obtained from Taylor expansions has improved local errors, but is worse in terms of error-propagation for long times or large time-steps. A first-order fully implicit method has similar errors, although it would be more stable for cases of stiff differential equations. The smallest errors were found with the semi-implicit or central difference algorithm.

2. Stochastic Equations

We consider a general multiplicative stochastic differential equation in N variables, of the structure

$$\frac{d}{dt} x_i(t) = a_i(t, \mathbf{x}) + \sum_j b_{ij}(t, \mathbf{x})\, \xi_j(t), \tag{2.1}$$

where

$$\langle \xi_i(t)\, \xi_j(t') \rangle_\infty = \delta_{ij} \delta(t - t').$$

Here $\xi_j(t)$ is a delta-correlated real Gaussian process with zero mean, and x_i is a real or complex variable. The equation has a wideband noise source, which is typical of stochastic differential equations obtained from a Fokker–Planck or similar equation. The notation $\langle\ \rangle_\infty$ is used here to denote an average over an infinite population of the random sources ξ_j. We use this notation to distinguish theoretical infinite population averages from the finite population averages used in numerical simulations. (In statistics the notation of a hat "$\wedge$" is sometimes used to indicate the statistics of a finite sample.)

The case of Lorentzian noise can also be treated in this general way. In this case, the noise sources have extra equations of the above form. The solutions to the additional equations have a Lorentzian spectrum. These then occur as source terms in the physical equations. Thus Eq. (2.1) can be used to treat a wide variety of interesting problems. It is applicable to both wideband and finite bandwidth noise sources.

Equation (2.1) can be regarded either as an Ito [2] or as a Stratonovich [3] type of stochastic equation. These are both commonly found in applications. A Stratonovich equation is typically obtained as the physical wideband limit of a finite bandwidth equation. Equation (2.1) is therefore taken in this paper as a Stratonovich equation, as this is often more relevant in statistical physics. Nevertheless, an Ito equation can always be calculated from Eq. (2.1). An Ito

process is defined as the limit of a sequence where the multiplicative term is evaluated at the start of any finite time interval and can be written as

$$\frac{d}{dt} x_i(t) = a_i^I(t, \mathbf{x}) + \sum_j b_{ij}(t, \mathbf{x})\, \xi_j^I(t). \tag{2.2}$$

Here the I indicates an Ito process; the process equivalent to Eq. (2.1) has

$$a_i^I(t, \mathbf{x}) = a_i(t, \mathbf{x}) + \frac{1}{2} \sum_j \sum_k b_{kj}(t, \mathbf{x}) \frac{\partial}{\partial x_k} b_{ij}(t, \mathbf{x}). \tag{2.3}$$

We note the presence of an extra term in the drift, called the Ito term, which compensates for the non-anticipating nature of Ito stochastic processes.

As an example of a Stratonovich equation, we consider the case of the Kubo oscillator [6] with a stochastic frequency. This has a typical equation of the form:

$$\frac{\partial}{\partial t} x(t) = ix(t)[\xi(t) + \omega_0]. \tag{2.4}$$

Since the equation is the wideband case of a physical oscillator, it must be regarded as a Stratonovich equation [1]. For this reason, ordinary calculus techniques of variable changes can be used in solving this equation. We note that Kubo oscillator equations of arbitrary noise strength can be written in this form, by use of a rescaling in (t).

Equation (2.4) has a direct solution on taking logarithms:

$$\log(x(t)/x(t_0)) = i \int_{t_0}^{t} [\xi(t') + \omega_0]\, dt'. \tag{2.5a}$$

Here $\xi(t)$ is a Gaussian process, and the result can be iterated over finite time intervals $\Delta t_n = t_{n+1} - t_n$ to give

$$\log[x(t_{n+1})/x(t_n)] = i[\omega_0 \Delta t_n + \Delta W^{(n)}], \tag{2.5b}$$

where

$$\langle \Delta W^{(n)} \Delta W^{(n')} \rangle_\infty = \Delta t_n \delta_{nn}$$

$$\Delta W^{(n)} = \int_{t_n}^{t_{n+1}} \xi(t)\, dt.$$

Since $\Delta W^{(n)}$ is a real Gaussian random variable of zero mean and known variance, it can be readily simulated numerically. The exact solution for $x(t_n)$ at any point is obtained on simply iterating Eq. (2.5b) over successive time intervals, for each given random sequence $\Delta W^{(n)}$, as in Fig. 1. Here we set $\omega_0 = 1$. The Gaussian random numbers are generated using a standard numerical subroutine, distributed by NAG (Numerical Algorithms Group), which implements Brent's algorithm [7].

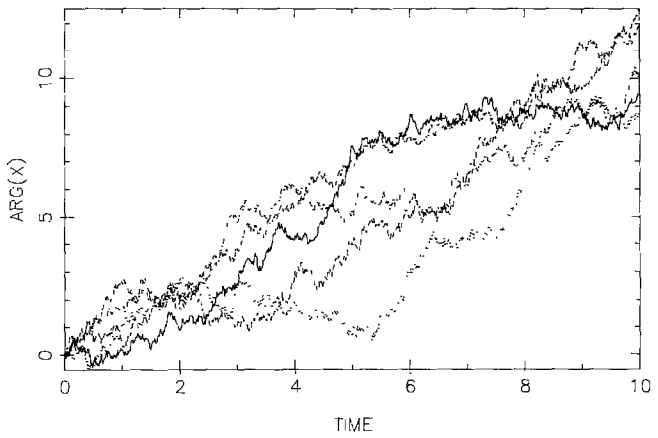

FIG.1. Exact solutions for the phase of a Kubo oscillator, with $\omega_0 = 1$, $\Delta t = 0.01$.

It is straightforward to demonstrate from this result that the value of any stochastic moment $\langle x^m \rangle_\infty$ over all possible trajectories $x(t)$, is given by

$$\langle [x(t)]^m \rangle_\infty = \langle [x(t_0)]^m \rangle_\infty e^{[im\omega_0 - m^2/2]t}. \tag{2.6}$$

Hence any moment of any order is calculable over an infinite population of stochastic trajectories. All the moments decay to zero exponentially, with a rate dependant on the order of the moment. Thus we have analytic results describing either individual stochastic trajectories or moments of an infinite number of stochastic trajectories. We note, of course, that individual trajectories have a constant modulus, while the average amplitude has a modulus that decays to zero.

A more general case, also of the type of Eq. (2.1), is the Kubo oscillator with finite bandwidth noise source [8]. In the case of Lorentzian noise, this can be written as a pair of equations:

$$\begin{aligned} \frac{\partial}{\partial t} x(t) &= ix(t)\,\omega(t) \\ \frac{\partial}{\partial t} \omega(t) &= k[\xi(t) - \omega(t) + \omega_0]. \end{aligned} \tag{2.7}$$

This set of equations still has the form of Eq. (2.1), which is therefore suitable for either finite or wide bandwith noise.

The solution for $\omega(t)$ over a finite time interval is

$$\omega(t) = e^{-kt}\left[e^{kt_0}\omega(t_0) + k \int_{t_0}^{t} e^{kt'}[\omega_0 + \xi(t')]\, dt' \right]. \tag{2.8a}$$

Hence the corresponding solution for $x(t)$ can be written as

$$\log(x(t)/x(t_0)) = i\int_{t_0}^{t} \omega(t_0)\, e^{-k(t'-t_0)}\, dt' + ik\int_{t_0}^{t}\int_{t_0}^{t'} e^{-k(t'-t'')}(\omega_0 + \xi(t''))\, dt''\, dt'. \tag{2.8b}$$

This equation shows that even cases of coloured noise, or equivalently, systems of stochastic differential equations, can also be treated exactly.

In general it is much more difficult to obtain solutions of stochastic equations in closed form when there are nonlinearities present, although some exactly soluble cases exist. In most cases, numerical approximations need to be introduced, giving rise to algorithms that are valid only in the limit of $\Delta t_n \to 0$. One procedure for numerically solving these equations is based on the definition of the Ito equation, since this can be defined as the limit of an Euler [9] algorithm for small step size. Let $\mathbf{x}^{(n)}$ be defined at discrete times t_n. The resulting Ito–Euler algorithm is known to have strong convergence to order $(\Delta t)^{1/2}$ in the sense of mean square error, and to have weak convergence to order (Δt) in the sense of moment generation [10, 11]. We therefore term this method a weak explicit method:

$$\text{Weak explicit:}\quad \Delta x_i^{(n)} = a_i^I(t_n, \mathbf{x}^{(n)})\,\Delta t_n + \sum_j b_{ij}(t_n, \mathbf{x}^{(n)})\,\Delta W_j^{(n)}, \tag{2.9}$$

where

$$\Delta x_i^{(n)} = x_i^{(n+1)} - x_i^{(n)}$$
$$\Delta t_n = t_{n+1} - t_n.$$

Here $\Delta W^{(n)}$ is a real Gaussian random variable with zero mean and variance Δt_n, as before. The solution to Eq. (2.2) is recovered on letting $\Delta t_n \to 0$. We show later that the Stratonovich process of Eq. (2.1) can be defined as in Eq. (2.9), with $\Delta t_n \to 0$, and with the functions a, b defined implicitly at $\bar{\mathbf{x}} = 1/2[\mathbf{x}^{(n)} + \mathbf{x}^{(n+1)}]$. The numerical implications of this definition will also be treated.

A careful simulation of Eq. (2.4) including the correction term of Eq. (2.3) leads to the results graphed in Fig. 2. The actual algorithm graphed is the Ito–Euler equation corresponding to Eq. (2.4). Thus, the technique used is

$$x^{(n+1)} = (1 - \tfrac{1}{2}\Delta t)\, x^{(n)} + ix^{(n)}[\Delta W^{(n)} + \omega_0\, \Delta t]. \tag{2.10}$$

In these graphs the results are compared by calculating the error relative to exact stochastic trajectories generated from identical underlying noise sources. With this procedure, the finite step size or discretization error can be readily distinguished from the finite population or sampling error. To allow direct comparisons at two

different step sizes, results with a step size of $2\,\Delta t$ are obtained from those with a step size of Δt by adding pairs of adjacent $\Delta W^{(n)}$ values. Thus the stochastic processes can be compared directly in all cases. This comparison method is more reliable in computing step size effects than calculating mean values, which generally leads to much greater dispersion. In other words, we plot the mean error, rather than the error of the mean, in these graphs. Plots 2a and 2b are numerical averages over finite populations.

It is clear from Fig. 2 that the global or longtime errors due to the finite stepsize reduce with smaller values of Δt. This numerical work demonstrates that the Ito–Euler algorithm can be used in the white-noise limit, provided the step sizes are very small. The numerical studies leading to Fig. 2 are in agreement with the known properties of the Kubo oscillator shown in Fig. 1. These results appear to be at variance with recent claims [8] that the Ito–Euler algorithm is inapplicable to the Kubo oscillator. Unlike these earlier simulations, there is no rapid decay here in individual Kubo oscillator amplitudes, although there is a decay in the average amplitude.

A further demonstration of the correctness of the Ito–Euler method is obtained on analytically calculating the mean value over an infinite population of trajectories, and comparing it to Eq. (2.6). The mean squared error can also be calculated in a similar way. We find that, after n steps, with $x^{(0)} = 1$,

$$\langle x^{(n)} \rangle_\infty = (1 + (i\omega_0 - 1/2)\,\Delta t)^n \tag{2.11}$$

and

$$\langle |x^{(n)} - x(t_n)|^2 \rangle_\infty = 1 + (1 + \tfrac{5}{4}\,\Delta t^2)^n - 2\,\mathrm{Re}((1 + (i\omega_0 + \tfrac{1}{2})\,\Delta t)^n\, e^{-n(i\omega_0 + 1/2)\,\Delta t}).$$

This expression for the mean agrees precisely with Eq. (2.6) on taking the limit of $\Delta t \to 0$ and $n \to \infty$, so that $n\,\Delta t = t$. The mean squared error also clearly vanishes in this limit, showing that the algorithm can reproduce individual trajectories as well as the mean value of the amplitude.

In Figs 2c and 2d we compare the results of using Eq. (2.10) over finite samples, with analytic results for the mean square error in the infinite sample limit. It is clear that the analytic discretization error predictions of Eq. (2.11) are subject to a sampling error which varies approximately as $1/\sqrt{N}$, where N is the number of stochastic trajectories that are averaged. This behaviour can be regarded as generic to stochastic problems. Ito–Euler algorithm results will be treated in greater detail in later sections.

For comparison purposes, the solution was also computed with a time-reversed Ito algorithm. This is obtained simply by reversing the time-direction of the algorithm in Eq. (2.10) and applying it to the time-reversed equation for the Ito oscillator. We note that the Kubo oscillator equation is a Stratonovich equation,

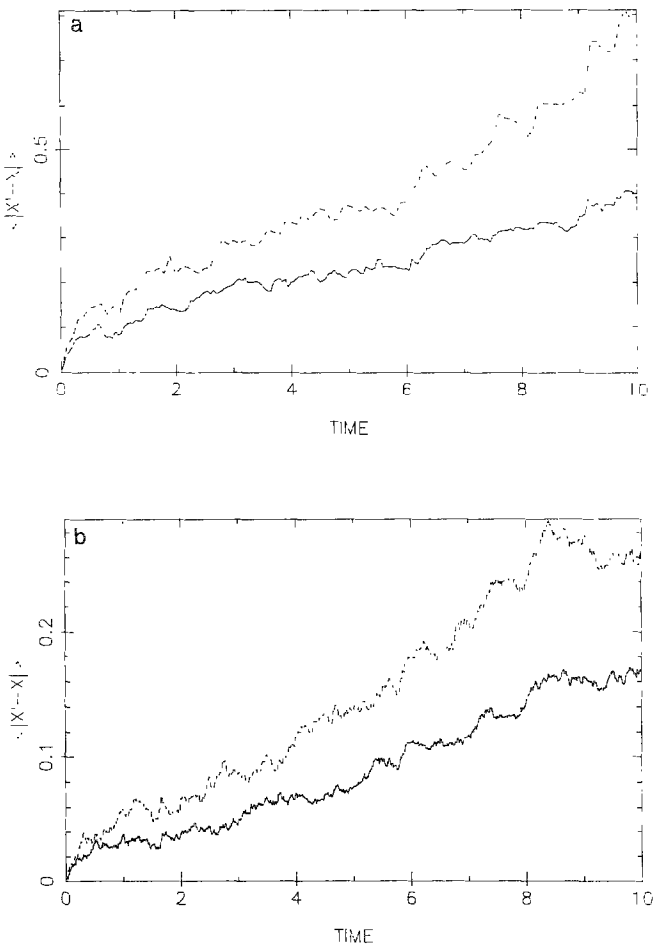

FIG. 2. Average error, $\langle |x' - x| \rangle$, where x' is the weak explicit or Ito–Euler solution of Eq. (2.11). Graphs show $\langle |x' - x| \rangle$ for 10 trajectories with $\omega_0 = 1$, $\Delta t = 0.1$, 0.05 (Fig. 2a) and $\Delta t = 0.01$, 0.005 (Fig. 2b). Figures 2c and 2d show the mean squared error in the mean compared to the exact, infinite sample result. Figure 2c has 100 and 10,000 trajectories with $\Delta t = 0.1$. Figure 2d has 100 and 10,000 trajectories with $\Delta t = 0.01$. In each case, the solid line is the predicted error in the limit of an infinite set of trajectories.

and so follows the usual calculus rules in time-reversal. The resulting algorithm is as follows:

$$x^{(n+1)} = [1 - \tfrac{1}{2}\Delta t - i(\omega_0 \Delta t + \Delta W^{(n)})]^{-1} x^{(n)}. \tag{2.12}$$

A thorough treatment of these types of implicit algorithm is given in Section 5, where we show this is a special case of a general weak implicit method. The results

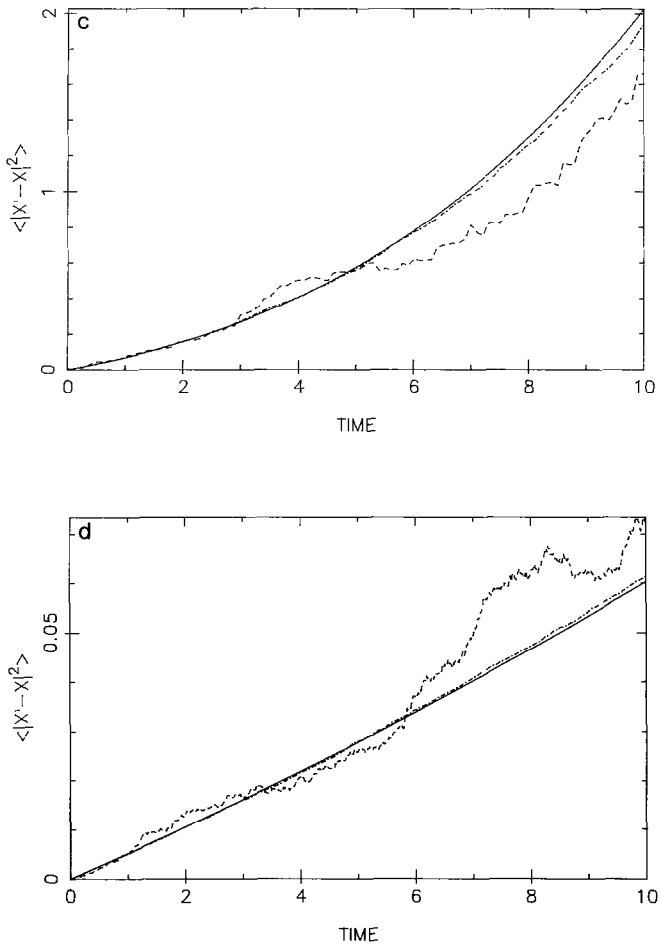

FIG. 2—*Continued.*

of using Eq. (2.12) are shown as the dotted curve in Fig. 3, where it can be seen that the errors are comparable to the normal Ito algorithm (solid curve).

It is interesting to note the close relationship between the discretization error and the Ito correction term. A Kubo oscillator has a circular trajectory on the complex plane, and a direct Euler simulation of the Stratonovich equation is always tangential to the circle of motion. This leads to a trajectory with rapidly increasing radius. Since the random source becomes relatively larger than Δt as $\Delta t \to 0$, the uncorrected error term does not vanish in the limit of $\Delta t \to 0$, when integrated over a finite time. Instead, it must be compensated for by the Ito correction term of

Eq. (2.3). This cancels, on average, the otherwise growing amplitude of the trajectory. Thus the Ito–Euler or weak explicit method is stochastically correct only to order ($\Delta W \sim \Delta t^{1/2}$) in the strong sense of individual trajectories, giving an overall accuracy to order (Δt, ΔW) in our notation.

3. Taylor Expansion Methods

While the Ito–Euler procedure is mathematically correct, it is numerically inefficient. Even worse, Euler procedures are numerically unstable [9] unless extremely small step sizes are used. This lack of robustness leads to increased discretization error and long computation times. For this reason, a number of alternatives to the Ito–Euler method have been suggested. These are reviewed by Kloeden and Platen [11]. One type of alternative is a generalised Runge–Kutta [12–14] algorithm. These algorithms are of higher order than the Ito–Euler one. Explicit Runge–Kutta methods, however, can lack robustness in their deterministic error-propagation properties, especially for stiff differential equations [15]. Despite this, these schemes give excellent local error performance, although it is known that the maximum order possible is limited.

Another proposed alternative is an explicit method like the Ito–Euler algorithm, except with stochastic correction terms [16–18] that replace the Ito correction. This technique is essentially an implementation of a stochastic Taylor expansion [19]. In this paper, we intend to treat algorithms suitable for multidimensional cases. Stability properties are therefore of especial importance, due to the possibility of stiff equations. For this reason, both implicit and explicit algorithms are treated. These algorithms are generally of higher stochastic order than the Ito–Euler algorithms, i.e., typically of order (Δt, ΔW^2). They also prove to have much smaller errors than the Ito–Euler method for higher-order moments, even though the Ito–Euler method is known to converge to order (Δt) in the weak sense of moment generation. The error-reduction is due to the fact that the stochastic error terms of $O(\Delta W^2)$ in the Ito–Euler algorithm can lead to large discretization errors in higher moments, which do not vanish when averaged over the trajectories. We note that in calculating moments numerically, there are also sampling errors due to finite populations. These are largely invariant with regard to the algorithm, so that we shall focus on the discretization error in most of the numerical results.

The algorithms treated in this paper are obtained directly using the Stratonovich equation, rather than going through a transformation to the Ito form. This allows the use of ordinary calculus to compute the results. It is also possible to produce implicit algorithms. These are known to have generally increased deterministic robustness when compared to either Euler *or* Runge–Kutta methods. The question of stability is especially important for stochastic equations, as the trajectories eventually map out all regions of the equation's phase space, including any region of numerical stiffness. In these regions an unstable algorithm can diverge, leading to incorrect or divergent global averages over all trajectories. Here we use the

deterministic terminology, although in stochastic equations Lyapunov exponents should be used to define stability properties.

It is therefore essential to choose numerical methods which are relatively stable in all regions of phase space. In particular, when there are nonlinearities in the coefficients **a**, **b**, the step size requirements for explicit methods can vary widely throughout phase-space. It is likely that for some equations there is in fact no finite step size that allows stable behaviour of the discretization error throughout phase space. This can lead to even more stringent requirements on **a**, **b** than those for the existence and uniqueness of the stochastic equations [1, 12]. The problem is made more severe by the fact that the stochastic terms vanish only as $(\Delta t)^{1/2}$ as $\Delta t \to 0$, so that these terms become relatively larger as $\Delta t \to 0$.

In order to overcome these problems, an analytic solution to Eq. (2.1) is obtained here which is approximately valid in each time interval (t_n, t_{n+1}). The first step is to linearize the coefficients **a**, **b**, as functions of **x**. A Taylor expansion in **x** around an intermediate point $\bar{\mathbf{x}}^{(n)}$ between $\mathbf{x}^{(n)}$ and $\mathbf{x}^{(n+1)}$ is calculated, where $\bar{\mathbf{x}}^{(n)} = \varepsilon \mathbf{x}^{(n+1)} + (1-\varepsilon)\,\mathbf{x}^{(n)}$. Hence, defining $\mathbf{z}^{(n)} = \mathbf{x} - \bar{\mathbf{x}}^{(n)}$,

$$\frac{\partial}{\partial t} z_i^{(n)}(t) = \bar{f}_i(t) + \sum_k \bar{g}_{ik}(t)\, z_k^{(n)}(t), \tag{3.1}$$

where

$$\bar{f}_i(t) = a_i(t, \bar{\mathbf{x}}) + \sum_j b_{ij}(t, \bar{\mathbf{x}})\, \xi_j(t)$$

$$\bar{g}_{ik}(t) = \frac{\partial}{\partial x_k}\left[a_i(t, \mathbf{x}) + \sum_j b_{ij}(t, \mathbf{x})\, \xi_j(t) \right]\Bigg|_{\mathbf{x} = \bar{\mathbf{x}}}.$$

Letting $\Delta \mathbf{x}^{(n)} = \mathbf{x}^{(n+1)} - \mathbf{x}^{(n)}$ as before, it is clear that $\mathbf{z}^{(n)}(t_n) = -\varepsilon\, \Delta \mathbf{x}^{(n)}$ and $\mathbf{z}^{(n)}(t_{n+1}) = (1-\varepsilon)\, \Delta \mathbf{x}^{(n)}$. The solution to Eq. (3.1) is immediate and gives the result,

$$(1-\varepsilon)\, \Delta x_i^{(n)} = \sum_j \left\{ \int_{t_n}^{t_{n+1}} U_{ij}(t_{n+1}, t)\, \bar{f}_j(t)\, dt - \varepsilon U_{ij}(t_{n+1}, t_n)\, \Delta x_j^{(n)} \right\}, \tag{3.2}$$

where

$$\frac{\partial}{\partial t} U_{ik}(t, t'') = \sum_j \bar{g}_{ij}(t)\, U_{jk}(t, t'')$$

$$U_{ik}(t, t) = \delta_{ik}.$$

This formal technique still requires that we obtain the solution of Eq. (3.2) which involves solving for U, and then integrating. In some cases, Eq. (3.2) can be treated exactly. Thus, in the case of a linear stochastic equation, an exact solution of Eq. (3.2) generates a numerical algorithm without any truncation error to all orders in (Δt). Approximate techniques for more general cases using stochastic Taylor

expansions will be given later. These calculations will only be taken here to first order in the Taylor expansion, for simplicity. However, a higher order expansion is also possible [19].

An exactly soluble example is the Kubo oscillator of Eq. (2.5). Here,

$$\begin{aligned} \bar{f}(t) &= i\bar{x}(\xi(t)+\omega_0) \\ \bar{g}(t) &= i(\xi(t)+\omega_0). \end{aligned} \tag{3.3}$$

The solution for $U(t, t_0)$ is then:

$$U(t, t_0) = \exp i \int_{t_0}^{t} (\omega_0 + \xi(t'))\, dt'.$$

The overall solution for $x^{(n+1)}$ given $x^{(n)}$ is independent of $\bar{x}$. It is obtained most readily on setting $\varepsilon = 0$, which gives

$$\begin{aligned} x^{(n+1)} &= \left[\exp i \int_{t_n}^{t_{n+1}} (\omega_0 + \xi(t))\, dt \right] x^{(n)} \\ &= e^{i(\Delta W^{(n)} + \omega_0 \Delta t_n)} x^{(n)}. \end{aligned} \tag{3.4}$$

Here $\Delta W^{(n)}$ is a Gaussian random number with variance Δt_n as in Eq. (2.10).

This result agrees precisely with the earlier analytic solution. Even though the random process $x(t)$ is known only at discrete sampling times t_j, the values obtained correspond to full solutions of the original stochastic equation. The results are accurate to all orders in (Δt), apart from possible numerical round-off errors. Thus, for example, all trajectories calculated this way have $|x^{(n)}| = 1$ if $|x^{(1)}| = 1$ initially. By comparison, the Ito–Euler simulations of Fig. 2 can only maintain the modulus $|x^{(n)}| = 1$ in the mean over short time intervals. For any step size Δt, the Ito–Euler algorithm generally results in trajectories with a slowly growing modulus over sufficiently long times. It is clear that at least for linear equations, it is possible to use the method of Eq. (3.2) to generate numerical trajectories of much greater accuracy than the Ito–Euler technique can offer. While the Kubo oscillator example is a relatively simple one, its exact numerical solution indicates the possibility of more powerful algorithms.

4. Explicit Algorithms

The formal result of Eq. (3.2) allows a large variety of numerical techniques to be generated in more general cases. These are obtained by different choices of the reference point $\bar{\mathbf{x}}$ and of methods of evaluating U_{ij}. If the choice $\varepsilon = 0$ is made with $\bar{\mathbf{x}} = \mathbf{x}^{(n)}$, an explicit method is obtained similar to the usual Ito–Euler algorithm, but with greater accuracy. We first rewrite f, g by expanding them about the

appropriate point in time $\bar{t}_n$, corresponding to $\bar{x}^{(n)}$, where $\bar{t}_n = \varepsilon t_{n+1} + (1-\varepsilon)\, t_n$ for the nth step. Thus, for the nth step,

$$\begin{aligned} \bar{f}_i(t) &= \bar{a}_i + \sum_j \bar{b}_{ij}\xi_j(t) + O(\Delta t) \\ \bar{g}_{ik}(t) &= \bar{a}_{ik} + \sum_j \bar{b}_{ijk}\xi_j(t) + O(\Delta t), \end{aligned} \tag{4.1}$$

where the coefficients $\bar{a}_i$, $\bar{a}_{ik}$, $\bar{b}_{ij}$, $\bar{b}_{ijk}$ are defined as

$$\begin{aligned} \bar{a}_i &= a_i(\bar{t}_n, \bar{\mathbf{x}}^{(n)}) \\ \bar{b}_{ij} &= b_{ij}(\bar{t}_n, \bar{\mathbf{x}}^{(n)}) \\ \bar{a}_{ik} &= \frac{\partial}{\partial x_k} a_i(\bar{t}_n, \mathbf{x})|_{\mathbf{x}=\bar{\mathbf{x}}^{(n)}} \\ \bar{b}_{ijk} &= \frac{\partial}{\partial x_k} b_{ij}(\bar{t}_n, \mathbf{x})|_{\mathbf{x}=\bar{\mathbf{x}}^{(n)}}. \end{aligned} \tag{4.2}$$

The solution then requires the calculation of U_{ik}, which is obtained from Eq. (3.2). This has the iterative solution:

$$U_{ik}(t_2, t_1) = \delta_{ik} + \int_{t_1}^{t_2} \bar{g}_{ik}(t)\, dt + \sum_j \int_{t_1}^{t_2} \left[\int_{t_1}^{t} \bar{g}_{ij}(t)\, \bar{g}_{jk}(t')\, dt' \right] dt + \cdots. \tag{4.3}$$

In order to collect terms in Eq. (3.2) to order (Δt), it is necessary to include expressions in $\xi(t)$ up to order ξ^2, since these occur in double integrals whose average value is of order (Δt). This requires that the first time integral in Eq. (4.3) is retained. Hence, to the lowest relevant order, Eq. (3.2) can be rewritten as

$$\begin{aligned} (1-\varepsilon)\, \Delta x_i^{(n)} = \sum_k \int_{t_n}^{t_{n+1}} \left[\delta_{ik} + \int_t^{t_{n+1}} \bar{g}_{ik}(t')\, dt' \right] \left[\bar{a}_k + \sum_j \bar{b}_{kj}\xi_j(t) \right] dt \\ - \varepsilon \sum_j \left[\delta_{ij} + \int_{t_n}^{t_{n+1}} \bar{g}_{ij}(t)\, dt \right] \Delta x_j^{(n)}. \end{aligned} \tag{4.4}$$

Next, this can be further simplified on dropping the last term, since $\varepsilon = 0$ for the explicit case,

$$\begin{aligned} \Delta x_i^{(n)} &= \bar{a}_i\, \Delta t_n + \sum_j \bar{b}_{ij}\, \Delta W_j^{(n)} \\ &\quad + \sum_j \sum_{j'} \bar{c}_{ijj'}\, \Delta W_{jj'}^{(n)} + O(\Delta t^{3/2}), \end{aligned} \tag{4.5}$$

where

$$\bar{c}_{ijj'} = \sum_k \bar{b}_{ij'k}\bar{b}_{kj}$$

and

$$\Delta W_{jj'}^{(n)} = \int_{t_n}^{t_{n+1}} \int_{t}^{t_{n+1}} \xi_j(t)\, \xi_{j'}(t')\, dt'\, dt.$$

Since the result is correct to order $(\Delta t, \Delta W^2)$, it is accurate to higher stochastic order than that given by the Ito–Euler algorithm. The explicit result of Eq. (4.5) was first obtained by Milstein [16] for an N-dimensional stochastic process. Rao *et al.* [17] have given a similar result accurate to $O(\Delta t^2)$. Due to the use of Stratonovich calculus, the present derivation is obtained simply using standard integration of the equations in a similar way to the derivation of Sancho *et al.* [18].

It is useful to calculate the mean value of the last term in Eq. (4.5), to understand its relationship to the Ito–Euler algorithm. Clearly, since $\xi_j(t)$ is uncorrelated with $\xi_k(t)$,

$$\left\langle \sum_j \sum_{j'} \bar{c}_{ijj'}\, \Delta W_{jj'}^{(n)} \right\rangle = \frac{1}{2} \Delta t_n \sum_j \sum_k b_{kj}(\bar{\mathbf{x}})\, \frac{\partial}{\partial x_k}\, b_{ij}(\mathbf{x})|_{\mathbf{x}=\bar{\mathbf{x}}}. \tag{4.6}$$

Thus the term in $\bar{c}_{ijj'}$ corresponds in the mean to the Ito correction term which would be included in a^I in the Ito stochastic process. This demonstrates that the Ito–Euler algorithm, which is of order $\Delta W \sim (\Delta t)^{1/2}$ for individual trajectories, does give mean values that are locally correct to order (Δt). This is expected in view of the known weak convergence to order (Δt) of the Ito–Euler method [10].

In order to utilise the improved equation, it is necessary to generate the time-ordered stochastic integrals $\Delta W_{jj'}^{(n)}$. These are correlated with $\Delta W_j^{(n)}$. For ease of computation $\Delta W_{jj'}^{(n)}$ is divided into a symmetric component and an antisymmetric component. The symmetric part is simplest to evaluate, since

$$\begin{aligned}[\Delta W_{jj'}^{(n)} + \Delta W_{j'j}^{(n)}] &= \int_{t_n}^{t_{n+1}} \int_{t_n}^{t_{n+1}} \xi_j(t)\, \xi_j(t')\, dt\, dt' \\ &= \Delta W_j^{(n)}\, \Delta W_{j'}^{(n)}.\end{aligned} \tag{4.7}$$

The antisymmetric part $A_{jj'}^{(n)}$ is uncorrelated with $\Delta W^{(n)}$ and can be evaluated numerically using random number generation techniques. The overall result for $\Delta W_{jj'}^{(n)}$ is that

$$\Delta W_{jj'}^{(n)} = \tfrac{1}{2}[\Delta W_j^{(n)}\, \Delta W_{j'}^{(n)} + A_{jj'}^{(n)}],$$

where

$$A_{jj'}^{(n)} = \int_{t_n}^{t_{n+1}} \int_{t}^{t_{n+1}} [\xi_j(t)\, \xi_{j'}(t') - \xi_j(t')\, \xi_{j'}(t)]\, dt'\, dt. \tag{4.8}$$

Techniques for the efficient generation of these antisymmetric time-ordered stochastic integrals are treated in Kloeden and Platen [11]. In many cases of

interest the matrix $c_{ijj'}$ is itself symmetric in j and j', as in the case of the Kubo oscillator. This is known as commutative stochastic noise [20, 21]. In these situations it is unnecessary to evaluate the antisymmetric stochastic term $A_{jj'}^{(n)}$. This implies that only the original random set $\Delta W_j^{(n)}$ with variance Δt_n is required at each time step. In the general case with an explicit algorithm ($\varepsilon = 0$) the following result is obtained [16], which is strongly convergent to order (Δt):

$$\text{Strong explicit:} \quad \Delta x_i^{(n)} = a(t_n, \mathbf{x}^{(n)})\,\Delta t_n + \sum_j b_{ij}(t_n, \mathbf{x}^{(n)})\,\Delta W_j^{(n)} + \frac{1}{2}\sum_j \sum_k c_{ijk}(t_n, \mathbf{x}^{(n)})(\Delta W_j^{(n)}\,\Delta W_k^{(n)} + A_{jk}^{(n)}). \tag{4.9}$$

With this algorithm, the normal Euler approximation to the Ito equation is replaced by an improved method that includes all stochastic terms of order (Δt). By comparison, the Ito–Euler method of Eq. (2.9) includes stochastic terms only to order (ΔW), together with a correction term which gives the mean derivative correctly to order (Δt). For the Kubo oscillator, it is clear that Eq. (4.9) reproduces the exact result of Eq. (3.4) correctly to order $(\Delta W)^2$, for each individual trajectory. However, Eq. (2.10) only reproduces individual trajectories to order (ΔW). The strong explicit algorithm in the Kubo oscillator case is

$$x^{(n+1)} = (1 + i(\Delta W^{(n)} + \omega_0\,\Delta t) - \tfrac{1}{2}(\Delta W^{(n)})^2)\,x^{(n)}. \tag{4.10}$$

The effect of the higher order correction is shown to be a relatively large one in Fig. 3, where errors are calculated by comparing individual trajectories in this algorithm with those in the exact result. It is obvious that Eq. (4.9) is of greater accuracy for small step sizes than Eq. (2.9). The error is, of course, zero in the exact algorithm of Eq. (3.4). Of more interest are the increased long-term errors in the calculation that are obtained with larger step sizes. These long-term errors are due to a faster growth in the Kubo oscillator amplitude in the Milstein algorithm, relative to the Ito–Euler case. This causes an increased relative error-propagation rate which is not immediately apparent from a local analysis of the truncation error.

In summary, the strong explicit algorithm of Eq. (4.9) corresponds to the Ito–Euler algorithm only in the mean. For each individual stochastic trajectory, this algorithm generates individual trajectories which are accurate to order (Δt). By comparison, the Ito–Euler method reproduces individual trajectories accurately only to order $(\Delta t)^{1/2}$ and neglects correction terms of order (Δt). These neglected correction terms have zero mean, but can obviously change the variances in an ensemble average. Thus the Milstein technique can reproduce higher order moments with increased accuracy relative to the Ito–Euler method. However, its global error-propagation properties are worse than those of the Ito–Euler algorithm for the Kubo oscillator, especially with larger step sizes. The strong explicit algorithm is therefore unsuitable for stiff equations, where error instabilities are a

serious problem. A theorem on global convergence properties with one-step approximations is proved by Milstein [22]. The Milstein theorem states that, under suitable conditions, an algorithm whose local mean square error is of order p_2 will produce a global mean square error of order p, where

$$p = p_2 - \tfrac{1}{2}. \tag{4.11}$$

This holds in the case that the stochastic equation coefficients (a, b) are constrained by the Lipschitz condition

$$|\mathbf{a}(t, \mathbf{x}) - \mathbf{a}(t, \mathbf{y})| + \sum_j |\mathbf{b}_j(t, x) - \mathbf{b}_j(t, \mathbf{y})| \leqslant K_1 |\mathbf{x} - \mathbf{y}| \tag{4.12}$$

where

$$\mathbf{b}_j = (b_{1j}, b_{2j}, b_{3j} \ldots).$$

While this is applicable to the Kubo oscillator case, we note that in physics applications the functions appearing often have faster than linear growth rates.

5. Implicit Algorithms

In this case of $\varepsilon > 0$, the calculation of each step requires knowledge of the endpoint, $\mathbf{x}^{(n+1)}$. This type of numerical method is known as an *implicit* one and generally requires the solution of a set of simultaneous equations at each step in time. Typically, such methods have greatly improved numerical stability, at the cost of some increase in the number of computations required. In the case of stochastic trajectories, there is already a relatively large overhead required to compute a set of random numbers $\Delta W_j^{(n)}$ at each point. The small extra cost of using an implicit method can therefore be justified in terms of improved accuracy and stability. Milstein [14] has also derived implicit schemes of a different type. Implicit deterministic schemes are used extensively [23].

By choosing $\varepsilon = 1$, a fully implicit technique is obtained with $\bar{\mathbf{x}}^{(n)} = \mathbf{x}^{(n+1)}$. These types of method are known to have excellent deterministic stability with respect to error propagation in cases of stiff differential equations [8]. For $\varepsilon = \frac{1}{2}$, the resulting semi-implicit methods correspond closely to the natural definition of a Stratonovich process. In this case $\bar{\mathbf{x}}^{(n)} = \frac{1}{2}[\mathbf{x}^{(n)} + \mathbf{x}^{(n+1)}]$, so the method is a central difference approximation. This algorithm is accurate to order $(\Delta t)^2$ in the low-noise limit and has a wide area of application.

In the fully implicit case, Eq. (3.2) can be rewritten as

$$\sum_j U_{ij}(t_{n+1}, t_n)\, \Delta x_j^{(n)} = \sum_j \int_{t_n}^{t_{n+1}} U_{ij}(t_{n+1}, t)\, \bar{f}_j(t)\, dt. \tag{5.1}$$

It is useful here to note that U_{ij} has the elementary multiplication properties:

$$U_{ik}(t_2, t_1) = \sum_j U_{ij}(t_2, t)\, U_{jk}(t, t_1) \qquad (t_2 \geqslant t \geqslant t_1). \tag{5.2}$$

Provided $U_{ij}(t_2, t_1)$ is invertible, its inverse can be written as $U_{ij}(t_1, t_2)$, so that Eq. (5.2) applies for all values of (t_1, t_2). This defines $\{U\}$ as a group for each particular stochastic trajectory and allows $\Delta x_j^{(n)}$ in the fully implicit case to be written as

$$\Delta x_i^{(n)} = \sum_j \int_{t_n}^{t_{n+1}} U_{ij}(t_n, t)\, \bar{f}_j(t)\, dt, \tag{5.3}$$

where $U_{ij}(t_n, t) \simeq \delta_{ij} - \int_{t_n}^{t} \bar{g}_{ij}(t')\, dt'$ $(t > t_n)$. Hence, a result similar to Eq. (4.9) is found, which is correct to order $(\Delta t, \Delta W^2)$:

$$\text{Strong implicit:} \quad \Delta x_i^{(n)} = a_i(t_{n+1}, \mathbf{x}^{(n+1)})\, \Delta t_n + \sum_j b_{ij}(t_{n+1}, \mathbf{x}^{(n+1)})\, \Delta W_j^{(n)} - \frac{1}{2} \sum_j \sum_k c_{ijk}(t_{n+1}, \mathbf{x}^{(n+1)})[\Delta W_j^{(n)}\, \Delta W_k^{(n)} - A_{jk}^{(n)}]. \tag{5.4}$$

We see that Eq. (5.4) is a time-reversed version of Eq. (4.9). In fact, this is a stochastic Taylor expansion around the future point $\mathbf{x}^{(n+1)}$. Note that the symmetric correction term changes sign, while the antisymmetric term A_{jk} has the identical sign to the explicit case.

Here, $\bar{a}$, $\bar{b}_{ij}$, and $\bar{c}_{ijk}$ must be evaluated at the future point $\mathbf{x}_i^{(n+1)}$, which generally requires the use of iterative techniques to compute the value of $\Delta \mathbf{x}$. As in the Milstein case, the algorithm is simplified when $\bar{c}_{ijk}$ is symmetric, giving commutative stochastic noise.

For the Kubo case, since the equation is linear, the numerical algorithm corresponding to Eq. (5.4) is

$$x^{(n+1)} = (1 - i(\Delta W^{(n)} + \omega_0\, \Delta t) - \tfrac{1}{2}(\Delta W^{(n)})^2)^{-1}\, x^{(n)}. \tag{5.5}$$

This is easily seen as corresponding to a binomial expansion of the inverse of the exponential in Eq. (3.4), where terms are collected to order $(\Delta W^{(n)})^2$. The results are shown in Fig. 3. This method has similar errors to the strong explicit algorithm at small stepsize, but is more stable in the case of larger stepsize.

It is also possible to define a time-reversed version of the Ito equation, by replacing the stochastic products in Eq. (5.4) by their mean value. This gives the following algorithm, which can be recognized as a time-reversed or implicit Ito–Euler method:

$$\text{Weak implicit:} \quad \Delta x_i^{(n)} = \left[a_i(t_{n+1}, \mathbf{x}^{(n+1)}) - \frac{1}{2} \sum_j c_{ijj}(t_{n+1}, \mathbf{x}^{(n+1)}) \right] \Delta t + \sum_j b_{ij}(t_{n+1}, \mathbf{x}^{(n+1)})\, \Delta W_j^{(n)}. \tag{5.6}$$

In the Kubo oscillator case, Eq. (5.6) generates Eq. (2.12), with numerical results that are shown in Fig. 3. Clearly, this behaves very similarly to the ordinary Ito–Euler method. As this method uses an implicit evaluation of the derivative, it can be expected to be more stable for the case of stiff equations, even though it is correct only to order $(\Delta t, \Delta W)$. The time-reversed or implicit Ito method does not appear to be mentioned widely in the literature. However, it is likely to have a wider domain of stability than most other methods currently in use.

Finally, suppose an intermediate point is utilised for the algorithm, with $\varepsilon = \frac{1}{2}$. In this case, Eq. (3.2) can be written as

$$\frac{1}{2}\sum_j (\delta_{ij} + U_{ij}(t_{n+1}, t_n)\,\Delta x_j^{(n)} = \sum_j \int_{t_n}^{t_{n+1}} U_{ij}(t_{n+1}, t)\,\bar{f}_j(t)\,dt. \tag{5.7}$$

This generates a semi-implicit method that is symmetric in time and strongly convergent to order (Δt). Going through a similar calculation to that already obtained gives the following result, correct to order $(\Delta t^2, \Delta W^2)$.

$$\text{Strong semi-implicit:}\quad \Delta x_i^{(n)} = a_i(\bar{t}_n, \bar{\mathbf{x}}^{(n)})\,\Delta t_n + \sum_j b_{ij}(\bar{t}_n, \bar{\mathbf{x}}^{(n)})\,\Delta W_j^{(n)} + \frac{1}{2}\sum_j \sum_k c_{ijk}(\bar{t}_n, \bar{\mathbf{x}}^{(n)})\,A_{jk}^{(n)}. \tag{5.8}$$

In this case the term in $\Delta W_j^{(n)}\,\Delta W_k^{(n)}$ is *identically* zero. In cases of commutative noise, the antisymmetric noise terms can give no contribution either. Noting that the antisymmetric terms $A_{jk}^{(n)}$ have zero mean, we see that a weak semi-implicit method is obtained by omitting these terms, even if the noise is non-commutative. This leaves the simple result

$$\text{Weak semi-implicit:}\quad \Delta x_i^{(n)} = a_i(\bar{t}_n, \bar{\mathbf{x}}^{(n)})\,\Delta t_n + \sum_j b_{ij}(\bar{t}_n, \bar{\mathbf{x}}^{(n)})\,\Delta W_j^{(n)}. \tag{5.9}$$

This algorithm now corresponds precisely to a standard implicit algorithm for solving differential equations. However, as usual, $\Delta W^{(n)}$ scales with $(\Delta t)^{1/2}$ instead of Δt. With regard to the terms in Δt only, the semi-implicit types of algorithm are of higher order than the earlier ones. They are correct to $O(\Delta t^2)$ in the limit of zero stochastic noise. More generally, even with non-commutative noise, Eq. (5.9) is still weakly convergent and therefore useful in generating moments, since the residual second-order noise term is of zero mean. This has been suggested as a stochastic method suitable for more general partial-differential space-time equations [24], as well as for stochastic time-domain problems [25, 26]. Thus, Eq. (5.9) generates an algorithm that is convergent to at least $O(\Delta t^2, \Delta W)$ in all cases, and to $O(\Delta t^2, \Delta W^2)$ in the case of commutative stochastic noise. In the non-commutative noise case, it can be expected to behave in a generally similar way to the Ito–Euler and time-reversed Ito–Euler methods, which are also correct to $O(\Delta W)$. However, it has a lower error in the deterministic limit.

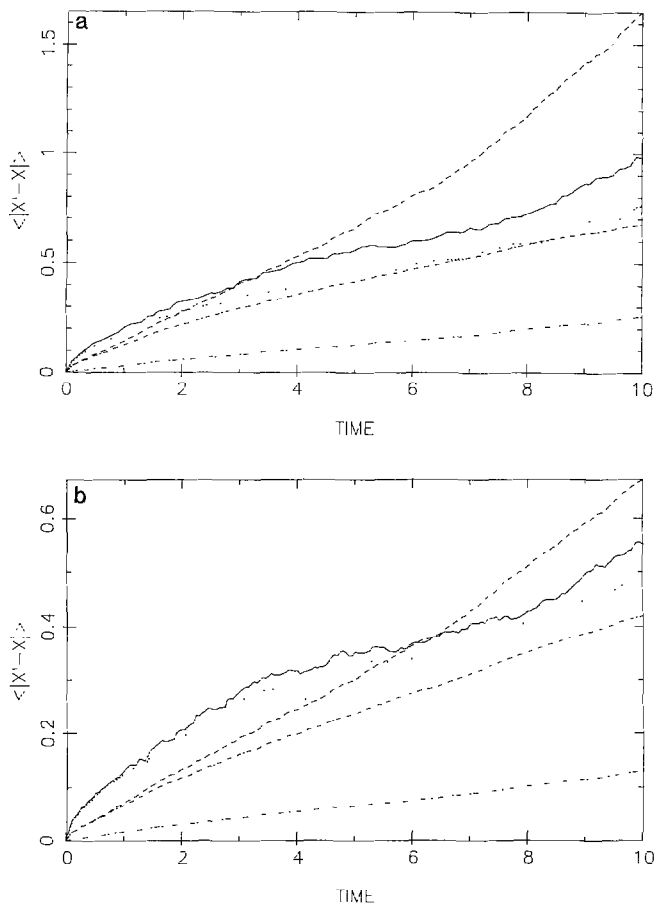

FIG. 3. Comparison of error performance of algorithms as in Fig. 2, for 100 trajectories. Figure 3a shows the error for $\Delta t = 0.1$. Figure 3b shows the error for $\Delta t = 0.05$, Fig 3c for $\Delta t = 0.01$, and Fig. 3d for $\Delta t = 0.005$. The key to the lines is as follows: ——— (weak explicit); ··· (weak implicit); – – (strong explicit); –·–· (strong implicit); –···– (semi-implicit).

In order to show how to utilise Eq. (5.9), consider an elementary Newton type of root-searching method, where $\bar{a}_i$ and $\bar{b}_{ij}$ are expanded to first order in $\Delta \mathbf{x}^{(n)}$ around $\mathbf{x}^{(n)}$. In this case,

$$\Delta x_i^{(n)} = \sum_j [\mathbf{G}(\mathbf{x}^{(n)})]_{ij}^{-1} \left[a_j(\mathbf{x}^{(n)})\, \Delta t_n + \sum_k b_{jk}(\mathbf{x}^{(n)})\, \Delta W_k^{(n)} \right], \tag{5.10}$$

where

$$G_{ij}(\mathbf{x}^{(n)}) = \delta_{ij} - \frac{1}{2} \frac{\partial}{\partial x_j} \left[a_i(\mathbf{x})\, \Delta t_n + \sum_k b_{ik}(\mathbf{x})\, \Delta W_k^{(n)} \right] \Bigg|_{\mathbf{x} = \mathbf{x}^{(n)}}.$$

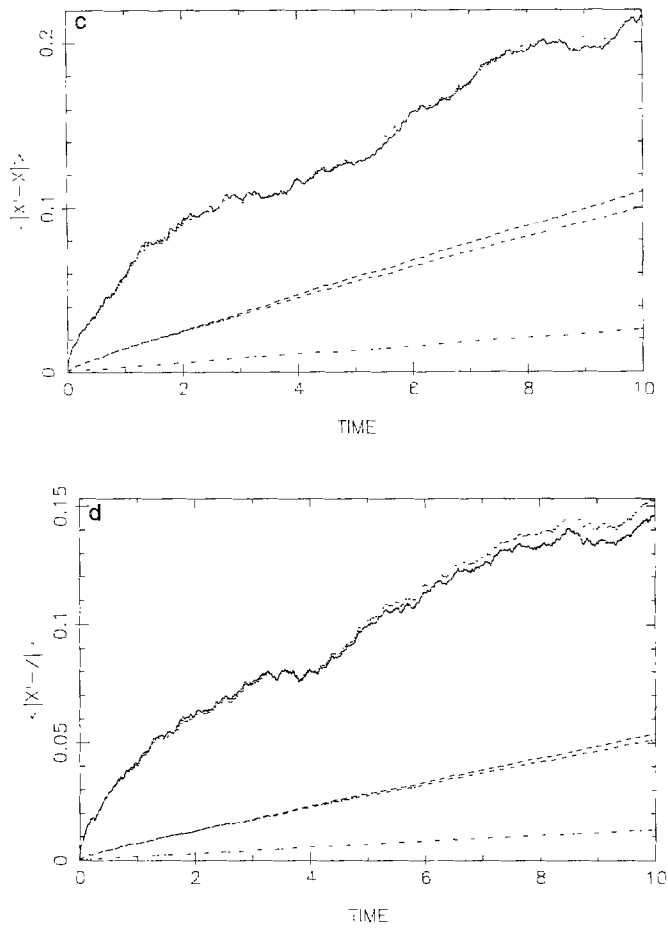

FIG. 3—*Continued*

This is exact in the case of the Kubo oscillator, which is linear in x. For comparison with Eq. (3.4), Eq. (5.10) can be rewritten for the Kubo oscillator as

$$x^{(n+1)} = \left[1 - \frac{i}{2}(\Delta W^{(n)} + \omega_0 \Delta t)\right]^{-1} \left[1 + \frac{i}{2}(\Delta W^{(n)} + \omega_0 \Delta t)\right] x^{(n)}. \quad (5.11)$$

This can also be obtained directly using Eq. (3.4). In fact, it can be readily verified that this expression agrees with the exact solution of Eq. (3.4) to all orders up to $(\Delta W^{(n)})^2$, but has a lower error than before to order $(\Delta W^{(n)})^3$. The third-order coefficient is 0.25, compared to the correct coefficient of 0.17 and a coefficient of 0.0 in the previous algorithms. The algorithm is also exact to order $(\Delta t)^2$ in the limit of low noise (i.e., ignoring the term in $\Delta W^{(n)}$). The results are of good accuracy and

stability, which is shown in Fig. 3. Both the short term and long term errors are greatly reduced in comparison with the other approximate algorithms.

More generally, for nonlinear equations, Eq. (5.10) can be improved by using iterative techniques which generate successive estimates of $\Delta \mathbf{x}^{(n+1)}$. A number of iterative procedures are known for this nonlinear root finding problem. Thus, in the Newton method, the μ-th estimate of $\Delta \mathbf{x}^{(n)}$ can be written $\Delta \mathbf{x}^{(n)}[\mu]$, where $\Delta \mathbf{x}^{(n)}[\mu+1]$ is obtained from $\Delta \mathbf{x}^{(n)}[\mu]$ on solution of the set of linear equations,

$$\sum_j \bar{G}_{ij}[\mu][\Delta x_j^{(n)}[\mu+1] - \Delta x_j^{(n)}[\mu]] = \bar{a}_i[\mu]\,\Delta t_n + \sum_k \bar{b}_{ik}[\mu]\,\Delta W_k^{(n)}, \quad (5.12)$$

where

$$\bar{G}_{ij}[\mu] = G_{ij}(\mathbf{x}^{(n)} + \tfrac{1}{2}\Delta \mathbf{x}^{(n)}[\mu])$$
$$\bar{a}_i[\mu] = a_i(\mathbf{x}^{(n)} + \tfrac{1}{2}\Delta \mathbf{x}^{(n)}[\mu])$$
$$\bar{b}_{ik}[\mu] = b_{ik}(\mathbf{x}^{(n)} + \tfrac{1}{2}\Delta \mathbf{x}^{(n)}[\mu]).$$

In cases of large computational complexity, $\bar{G}_{ij}$ can be replaced by its initial value at $\mathbf{x}^{(n)}$. Alternatively, the secant rule or quasi-Newton techniques [9] can be utilised for solving the implicit equations. This technique is also applicable, with appropriate adaptions, to Eq. (5.8). In general these techniques can modify the error bounds when a finite number of iterations are used. However, this is easily checked in practise by varying the number of iterations.

As a precautionary note, it is necessary for the implicit equations to have a solution in the neighbourhood of the previous point. In order for these methods to provide sensible results, the step size Δt must be reasonably small. Even with small Δt, the distribution of ΔW implies that large ΔW will occur with finite probability. For sufficiently large ΔW, it is likely that the inverse matrices will have a singularity. This implies that the Gaussian distribution of ΔW could need to be truncated for these methods. Of course, as Δt is reduced, the error caused by the truncation approaches zero rapidly, since the truncation points can be increased relative to the standard deviation. In the computer-generated results obtained here, no truncation was necessary.

6. Summary

A class of algorithms for the computer simulation of multiplicative stochastic processes is presented. The algorithms are based on an exact solution of the linearized equations over a short time interval, without using time-domain Taylor expansions. Instead, the expressions used here can be regarded as generalised stochastic Taylor expansions [27]. The parameter ε used in the algorithm defines the degree of implicitness. In fact, ε can be regarded as effectively interpolating between the usual Ito and Stratonovich type of differential equation, as well as

extending these to include a fully implicit or time-reversed Ito equation. The chief difference between these algorithms and the usual Ito–Euler type is that the stochastic corrections to order (Δt) are computed exactly for each trajectory, rather than just in the mean. We treat three algorithms that are only weakly convergent to $O(\Delta t)$, and three that converge strongly to $O(\Delta t)$, giving improved accuracy overall.

The higher-order algorithms are generally accurate only with small time-steps. The strong explicit or Milstein method can show worse long-term characteristics than the weak explicit or Ito–Euler algorithm due to error propagation, when the time-steps are relatively large. The implicit algorithms of either order are preferable

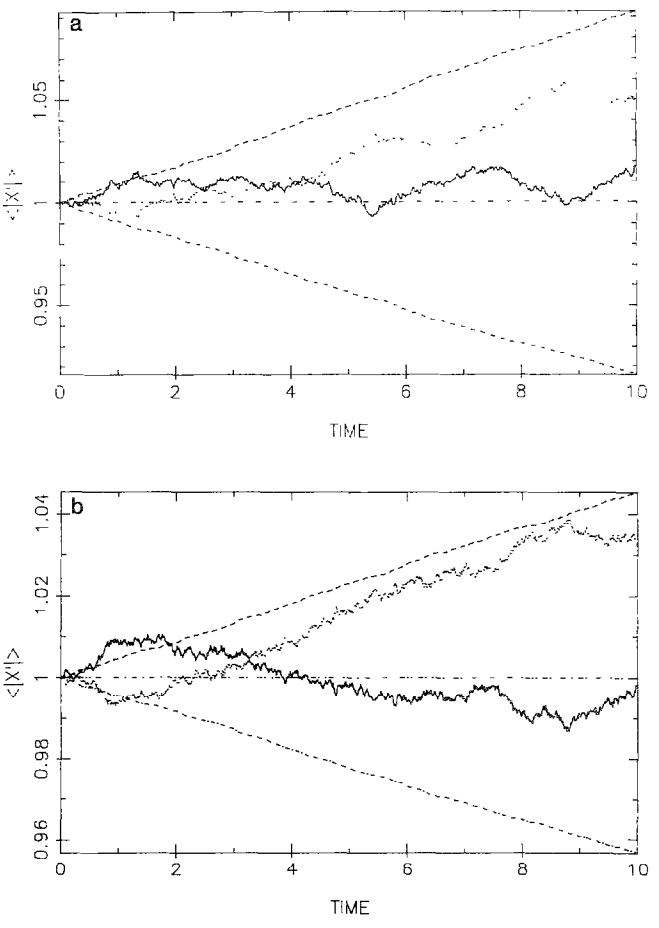

FIG. 4. Comparison of the calculated modulus $\langle |x'| \rangle$ for the five approximate algorithms Figure 4a has $\omega_0 = 1$, $\Delta t = 0.01$, with 100 trajectories. The key is as in Fig. 3. Figure 4b has identical parameters, except with $\Delta t = 0.005$.

for equations that are stiff or unstable when using explicit methods. The greatest accuracy in the Kubo oscillator case is obtained for the symmetric or semi-implicit case of Eq. (5.9). This case also corresponds to a second-order method in the deterministic limit and therefore is especially useful in cases of low noise. We note that other classes of algorithm or different stochastic equations could give rise to alternative optimal methods.

Detailed comparisons of the present algorithms are given in Figs. 3–6, for the case of the Kubo oscillator. We note that in this case the weak and strong methods of semi-implicit type are identical, as the stochastic noise is commutative. Here Fig. 3 graphs the mean error characteristics for four different step sizes, showing

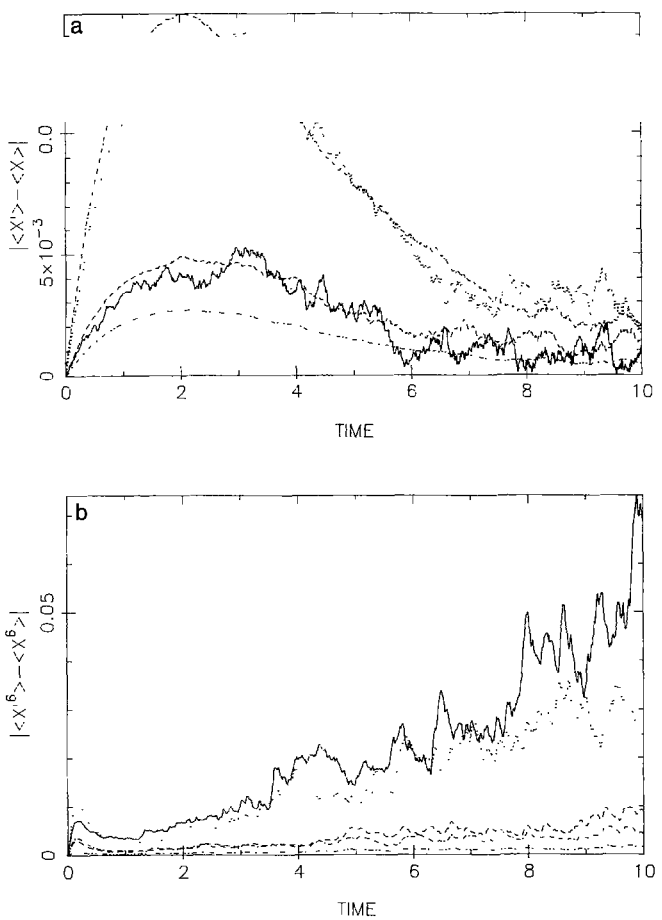

FIG. 5. Comparison of estimated moments for the algorithms. Plotted are the errors in the mean $|\langle x\rangle - \langle x'\rangle|$ in Fig. 5a and of the sixth moment, $|\langle x^6\rangle - \langle (x')^6\rangle|$, in Fig. 5b. Here $\omega_0 = 1$, $\Delta t = 0.01$, with 10,000 trajectories. Key as in Fig. 3. Figure 5b includes a time-domain smoothing of the graph over every 20 adjacent points, to improve visibility.

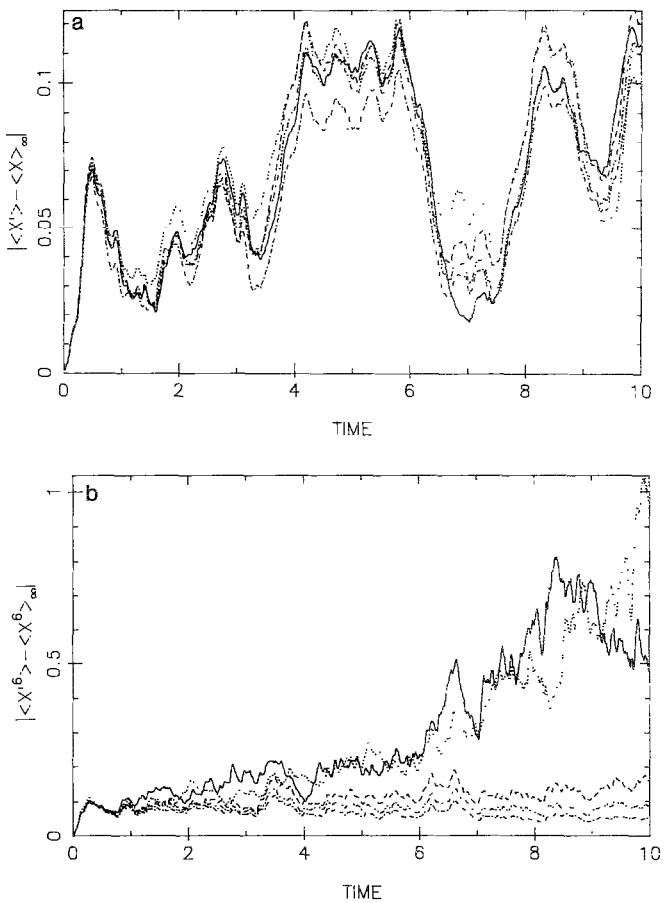

FIG. 6. In Fig. 6a and Fig. 6b $|\langle x' \rangle - \langle x \rangle_\infty|$ and $|\langle x' \rangle^6 - \langle x^6 \rangle_\infty|$ are plotted; i.e., they are the errors relative to the infinite sample result of Eq. (2.6). The combined truncation error and sampling error are shown for $\omega_0 = 1$, $\Delta t = 0.01$. These results are for 100 trajectories, while Figs. 6c and 6d show results for 10,000 trajectories. These figures include a time-domain smoothing over every 20 adjacent points, to improve visibility. The key to the lines is as in Fig. 3.

how the relative performance alters with step size. The ensemble used was of 100 distinct trajectories. For larger step sizes, the error-propagation problem is significant for all except the semi-implicit method. We note that the strong explicit or Milstein algorithm has the worst error-propagation rate for larger step sizes. Figure (4) shows the behaviour of the calculated modulus. None of the algorithms shows the rapid decay in the modulus which was obtained in an earlier simulation [8]. In addition, the symmetric or semi-implicit method has $|x| = 1$ always, which agrees with the exact result. Two-different step sizes are compared, with identical underlying noise sources, to demonstrate the reduction in error with reduced step size.

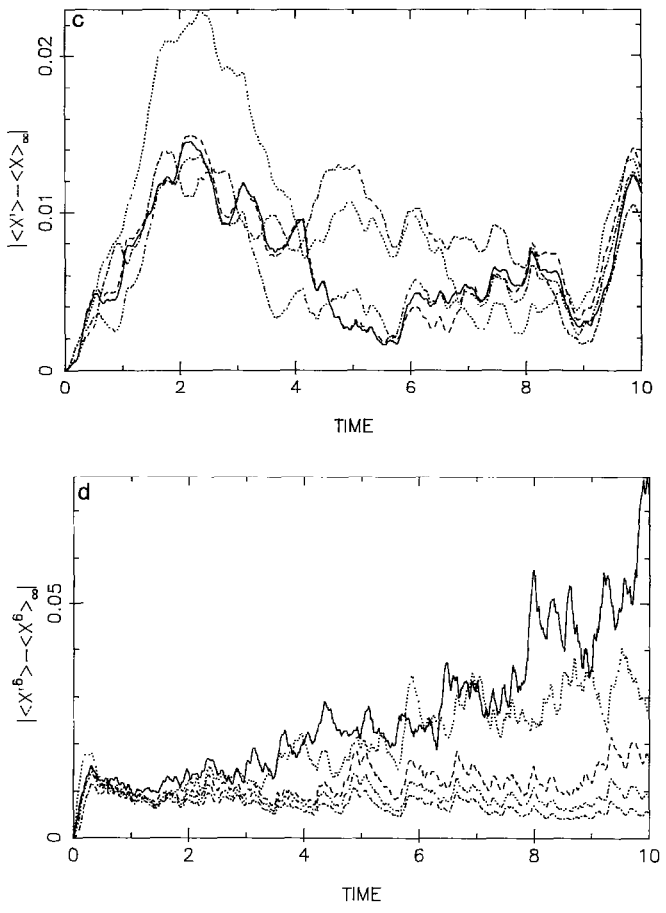

FIG. 6—*Continued*

Figures 5a and b show the error in the first- and sixth-order moments. Here the ensemble size had to be increased to 10,000 trajectories, to reduce the sampling error. Even so, the errors in the sixth-order moment showed large fluctuations, and the graphs are time-averaged over 20 neighbouring time-domain points to improve visibility. The results were checked in double precision (14 digit) arithmetic to ensure that round-off error was negligible. The weak explicit or Ito–Euler and weak implicit algorithms perform poorly for higher-order moments, since these methods have errors of order $(\Delta t)^{1/2}$ for individual trajectories, although the mean values are correct to order (Δt). Thus the errors in the mean are comparable with the other algorithms, but the higher order moments have much larger errors, for a given step size, even though they converge with the same order as the mean. In all cases treated, the semi-implicit algorithm has the smallest discretization error.

Calculating stochastic moments on a finite population necessarily results in additional random sampling errors relative to the ideal infinite population results. From the central limit theorem it follows that these random errors typically scale with $N^{-1/2}$ for N samples. As the error is approximately uniform for all algorithms or step sizes, we have graphed only the discretization errors so far, by comparing discretized and exact results over identical finite populations. However, the choice of step size needs to be combined with an appropriate choice of population size, in order to obtain optimal moment estimates. We therefore show the total error in the calculated moments, including sampling error, in Figs. 6a–d. These are obtained for 100 and 10,000 trajectories at a stepsize of $\Delta t = 0.01$. The results show that with this relatively small choice of step size, the mean value errors are largely dominated by the underlying sampling errors, which are similar for all algorithms. However, the sampling error in the higher order moments is strongly dependent on the algorithm chosen. The semi-implicit algorithm generally has the lowest total error, especially for the smaller populations. It is interesting to note that, for the higher order moments, the weakly convergent algorithms have much larger sampling errors than the strongly convergent methods.

In conclusion, the higher-order algorithms obtained here give much smaller errors than the Ito–Euler method, especially when calculating higher order moments. The strong implicit method is probably the most stable in the case of stiff equations. We note that it is also possible to obtain a weak implicit method using a time-reversed Ito–Euler algorithm. This also can be expected to have good stability properties. However, the semi-implicit method gives the least discretization error and the least sampling error for the Kubo oscillator. It therefore appears preferable for this problem.

ACKNOWLEDGMENTS

One of us (P.D.D.) acknowledges useful discussion with R. Roy and R. F. Fox We also acknowledge helpful suggestions from the referees.

REFERENCES

1. For general results on stochastic differential equations see, for example, L. ARNOLD. *Stochastic Differential Equations: Theory and Applications* (Wiley, New York, 1974); C. W. GARDINER, *Handbook of Stochastic Methods* (Springer-Verlag, Berlin, 1983).
2. K. ITO, *Lectures on Stochastic Processes* (Tata Institute of Fundamental Research, Bombay, 1960).
3. R. L. STRATONOVICH, *Topics in the Theory of Random Noise*, Vols. I, II (Gordon & Breach, New York, 1963).
4. E. HELFAND, *Bell Systems Tech. J.* **58**, 2289 (1979).
5. D. L. ERMAK AND H. BUCKHOLZ, *J. Comput. Phys.* **35**, 169 (1980).
6. R. KUBO, M. TODA, AND N. HASHITSUME, *Statistical Physics II* (Springer-Verlag, Berlin, 1985).
7. R. P. BRENT, *Commun. ACM.* **17**, 704 (1974).

8. R. F. Fox, R. Roy, A. W. Yu, *J. Statist. Phys.* **47**, 477 (1987). The spurious decays in the Kubo amplitude that these authors report were due to an error in implementing the numerical algorithm in the white-noise case. (R. F. Fox, private communication).
9. G. E. Forsythe, M. A. Malcolm, and C. B. Moler, *Computer Methods for Mathematical Computations* (Prentice–Hall, Englewood Cliffs, NJ, 1977); H. Risken, *The Fokker–Planck Equation* (Springer-Verlag, Berlin, 1984).
10. I. I. Gikhman and A. V. Skorokhod, *Stochastic Differential Equations* (Springer, Berlin, 1972).
11. P. E. Kloeden and E. Platen, *J. Stochastic Hydro. Hydraul.* **3**, 155 (1989).
12. W. Rumelin, *SIAM J. Numer. Anal.* **19**, 604 (1982); J. R. Klauder and W. P. Peterson, *SIAM J. Numer. Anal.* **22**, 1153 (1985).
13. D. Talay, *Springer Lecture Notes in Control and Information Science* **61**, 294 (1984); E. Pardoux and D. Talay, *Acta. Appl. Math.* **3**, 22 (1985).
14. G. N. Milstein, *Theor. Probab. Appl.* **30**, 750 (1986).
15. R. W. Hamming, *Numerical Methods for Scientists and Engineers* (McGraw–Hill, New York, 1962).
16. G. N. Milstein, *Theor. Probab. Appl.* **19**, 557 (1974).
17. N. J. Rao, J. D. Borwankar, and D. Ramkrishna, *SIAM J. Control* **12**, 124 (1974).
18. J. M. Sancho, M. San Miguel, S. L. Katz, J. D. Gunton, *Phys. Rev. A* **26**, 1589 (1982).
19. E. Platen, *Sankhyā A* **44**, 163 (1982).
20. H. Sussman, *Ann. Probab.* **6**, 19 (1978).
21. J. M. C. Clark and R. J. Cameron, *Springer Lect. Notes in Control and Information Science* **25**, 162 (1980).
22. G. N. Milstein, *Theor. Probab. Appl.* **32**, 738 (1987).
23. J. C. Butcher, *The Numerical Analysis of Ordinary Differential Equations. Runge–Kutta and General Linear Methods.* (Wiley, Chichester, UK, 1987).
24. P. D. Drummond, *Compt. Phys. Commun.* **29**, 211 (1983).
25. K. J. McNeil and I. J. D. Craig private communication (1989).
26. A. M. Smith and C. W. Gardiner, *Phys. Rev. A* **39**, 3511 (1989).
27. P. E. Kloeden and E. Platen, to appear.